A Provenance-Based Trust Model for Delay Tolerant Networks

Jin-Hee Cho, Moonjeong Chang, Ing-Ray Chen, Ananthram Swami

A Provenance-based Trust Model for Delay Tolerant Networks

Jin-Hee Cho[1], MoonJeong Chang[2], Ing-Ray Chen[2], Ananthram Swami[1]

[1] Computational and Information Sciences Directorate, U.S. Army Research Laboratory,
{jinhee.cho, ananthram.swami}@us.army.mil
[2] Department of Computer Science, Virginia Tech,
{mjjang, irchen}@vt.edu

Abstract. Managing trust efficiently and effectively is critical to facilitating cooperation or collaboration and decision making tasks in tactical networks while meeting system goals such as reliability, availability, or scalability. Delay tolerant networks are often encountered in military network environments where end-to-end connectivity is not guaranteed due to frequent disconnection or delay. This work proposes a provenance-based trust framework for efficiency in resource consumption as well as effectiveness in trust evaluation. Provenance refers to the history of ownership of a valued object or information. We adopt the concept of provenance in that trustworthiness of an information provider affects that of information, and vice-versa. The proposed trust framework takes a data-driven approach to reduce resource consumption in the presence of selfish or malicious nodes. This work adopts a model-based method to evaluate the proposed trust framework using Stochastic Petri Nets. The results show that the proposed trust framework achieves desirable accuracy of trust evaluation of nodes compared with an existing scheme while consuming significantly less communication overhead.

Keywords: delay tolerant network, provenance, store-and-forward, message carrier, trust, trustworthiness.

1 Introduction

Delay or disruption tolerant networks (DTNs) are often observed in emerging applications such as emergency response, special operations, smart environments, habitat monitoring, and vehicular ad hoc networks. The core characteristic of DTNs is that there is no guarantee of end-to-end connectivity, thus causing high delay or disruption due to various inherent characteristics (e.g., wireless medium, resource constraints, or high mobility) or intentionally misbehaving nodes (e.g., malicious or selfish) [13]. Due to the characteristics of DTNs, trust management techniques are vital for effectively and efficiently identifying untrustworthy nodes based on accurate trust evaluation and low network resource consumption. We propose a provenance-based trust model to achieve both goals.

The Institute for Information Infrastructure Protection (I3P) emphasized the importance of data provenance for secure, efficient, and trustworthy systems, as one

of the top homeland security research challenges in the 2009 report to the US Senate [18]. Data provenance has been used to analyze scientific data in many applications. The Open Provenance Model (OPM) was introduced to represent data provenance, process documentation, data derivation, and data annotation [10]. Since then, OPM has been widely adopted and extended by various research groups [8]. Freire et al. [5] surveyed various models of provenance management but did not discuss the use of provenance for security. McDaniel [9] associated security with provenance in that good security leads to good provenance with accurate, timely, and detailed provenance information, resulting in good security decisions.

Provenance has been used to verify trust, trustworthiness, or correctness of information in various research areas. Rajbhandari et al. [12] examined how provenance information is associated with a workflow in a Bio-Diversity application. Dai et al. [4] proposed a data provenance trust model to evaluate trustworthiness of data and data providers. Yu et al. [17] presented an agent-based approach to managing information trustworthiness in network centric information sharing environments. Golbeck [6] used provenance information to infer trust in Semantic Web-based social networks. However, the above works [4, 6, 12, 17] focused on evaluating trustworthiness in information without considering particular network attack models that may maliciously change the original messages and disrupt system goals.

Several provenance-based trust models have been proposed to evaluate trustworthiness of both sensed information and information providers (sensors) in sensor networks. Alam and Fahmy [1] proposed an energy-efficient provenance transmission and construction scheme for trust frameworks for evaluating trustworthiness of a sensed data item. Sultana [15] exploited the watermarking characteristics of their provenance mechanism to identify packet-dropping nodes. Wang et al. [16] and Lim et al. [7] proposed a provenance-based trust model to evaluate trust in information and sensors assuming that all paths are known and nodes are stationary. All the above works [1, 7, 15, 16] assumed full knowledge of the network topology, and did not consider attackers. Srivatsa et al. [14] exploited provenance information to propose an efficient cache strategy in DTNs, but did not consider attack behaviors.

In this work, we extend the existing provenance techniques for trust evaluation in DTNs; the challenges are due to the attackers who may modify or drop messages including provenance information or disseminate fake information. Leveraging the interdependency of trust in information and sources based on provenance, this work aims to achieve two goals for effective mission execution: (1) conducting accurate trust evaluation; and (2) incurring low communication overhead for trust evaluation.

We propose a provenance-based trust model that has the following features. First, the proposed scheme significantly reduces communication overhead by not incurring extra communication overhead for trust evaluation purposes in addition to message delivery. We achieve this by using provenance information (i.e., identification and opinion towards a previous message carrier) tagged in delivered message. In our protocol, a trustor does not directly request recommendations from third parties because collecting recommendations requires extra overhead, and recommendations are often not available in a sparse DTN. In addition, collecting indirect evidences via message delivery enables trust update even for two nodes that have not encountered each other for a long time. Second, we use reward and penalty strategies (i.e.,

increasing or decreasing trust level) to encourage nodes to behave. Third, our proposed trust model uses a composite trust metric embracing three trust properties: availability, integrity, and competence. Based on the literature [3], most existing trust management schemes evaluate a single trust property of a node in order to derive its trustworthiness. Last, we use a model-based evaluation method based on Stochastic Petri Nets (SPN) to identify an optimal minimum trust threshold (in selecting the[1]next message carrier) that maximizes trust accuracy while introducing low communication overhead.

2 System Model

We propose a distributed provenance-based trust management protocol. Each node is assumed to have capability to monitor its neighboring nodes with known probabilities of false positives and negatives in detecting attack behaviors or energy level.

2.1 Key Management

A node encrypts the entire "packet" (consisting of the message and provenance information) using a symmetric key $K_{S,t}$ given to legitimate members. Several trusted authorities (TAs) exist in the operational area so that a node is allowed to access a TA to obtain a valid symmetric key. However, the node may not be able to obtain a valid symmetric key either because no TA is available due the node's physical location or because its trust level is too low, below the minimum required system trust threshold T_{min}. TAs rekeys the symmetric key $K_{S,t}$ periodically based on their pre-deployed hash functions. The symmetric keys issued at the same time t by multiple TAs are the same so that all legitimate nodes can communicate with the same key. The symmetric key is used to prevent outside attackers, not inside attackers. A node forwards a packet to a node whose trust is equal to or above T_{min}.

We define the *provenance information* (PI) generated by node i as the tuple $(i, k, O_{i,k}^{D-integrity}(t))$, where k is the identification (ID) of the previous message carrier (MC), and $O_{i,k}^{D-integrity}(t)$ is the direct trust opinion of node i towards the previous MC k about its integrity. We use three attack behaviors to form the trust opinion: no identity or fake identity, mission message modification, and good/bad mouthing. We call a message to be used for mission execution as a "mission message (MM)" for notational convenience hereafter. Equation 4 describes how $O_{i,k}^{D-integrity}(t)$ is computed from its three trust components.

We simply denote $(i, k, O_{i,k}^{D-integrity}(t))$ as $P_{(i,k)}$ meaning PI provided by node i with its direct trust opinion towards the previous MC k. For example, a destination node (DN) may receive a message such as:

[1] In a typical MANET, one talks about the next hop node or the downstream or upstream neighbor. In a DTN, a node may carry a message for a long time until it encounters a node to whom this message can be passed on. We call this "next-hop node" as the next message carrier.

$$\left[\mathrm{MM}, \left(\mathrm{P}_{(0,\emptyset)}\right)_{\mathrm{k_n}}, \left(\mathrm{P}_{(1,0)}\right)_{\mathrm{k_{n-1}}}, \left(\mathrm{P}_{(2,1)}\right)_{\mathrm{k_{n-2}}}, \ldots, \left(\mathrm{P}_{(\mathrm{m,m-1})}\right)_{\mathrm{k_{n-m}}}\right]_{\mathrm{K_{S,t}}} \quad (1)$$

where MM denotes a mission message and $\mathrm{K_{S,t}}$ is a symmetric key issued at time t. The source node's ID is 0, and other intermediate MCs' IDs are 1, 2, … , m where m is the number of intermediate MCs. The message including both MM and PIs is encrypted by a symmetric key $\mathrm{K_{S,t}}$. Note that the source only encloses its ID since there is no previous MC. The apparent redundancy in the carried ID information is crucial in identifying some attacks, as discussed later. Typically, the addition of meta data by each relay node could lead to the so-called meta-data explosion problem if the number of hops or relays, m, is too large. However, this work does not have this problem because the proposed protocol is applied in a sparse DTN and it uses a trust threshold to filter trustworthy MCs.

To prevent modification of PIs inserted by previous MCs, we adopt an encryption key mechanism based on micro-TESLA [11]. Source and destination nodes obtain a base PI encryption key and decryption key, $(\mathrm{k_0}, \mathrm{k_n})$, from the closest TA. We assume that TAs are able to issue the same pair of keys (i.e., $(\mathrm{k_0}, \mathrm{k_n})$) to a pair of source and destination nodes. A source encrypts its PI using $\mathrm{k_n}$ and generates $\mathrm{k_{n-1}} = \mathrm{F}(\mathrm{k_n})$ to dictate the next MC to use $\mathrm{k_{n-1}}$. Similarly, the next MC will encrypt its PI using $\mathrm{k_{n-1}}$ and pass $\mathrm{k_{n-2}}$ to its next MC. This process continues until the message arrives at a DN. A MC does not know the previous MC's PI encryption key, so it cannot decrypt the PI of the previous MC. When the DN receives the message, it can check with $(\mathrm{k_0}, \mathrm{k_n})$ if correct keys are being used on the path, and can properly decrypt all PIs by tracing back the key chains.

Unless attackers capture the source or destination node, PIs cannot be fully altered. Attackers may collude and exchange PI encryption keys but PI modifications may occur between attackers themselves which have little impact on overall attack behaviors. If a MC does not comply with using a given PI encryption key, the DN will fail to decrypt all PIs and discard the message. This will eventually lead to identifying malicious nodes. Thus, we assume that smart attacker might want to follow the key policy to gain trust. However, using PI encryption/decryption keys does not guarantee that each MC provides correct provenance information. We consider that a node may drop or modify its own PI.

Symmetric keys and PI encryption/decryption keys are distributed via a public/private key pair. Each node will use a TA's public key to request proper keys and a TA is preloaded with public keys of all nodes in the network. Each node will decrypt a message carrying the symmetric or PI encryption key using its private key. Thus, non-TA nodes do not need to store public keys of all nodes. TAs are involved only in key management, not in the trust evaluation process.

2.2 Attack Model

The use of a symmetric key prevents outside attackers, but not inside attackers. We consider the following insider attacks:

- **Fake identity or no identity:** Our protocol requires that a MC should insert its ID in the PI tuple. However, an attacker may not add its real ID or may insert a

fake ID. If this attack is successful, this attacker's misbehavior may be interpreted as another node's misbehavior, leading to inaccurate trust evaluation.

- **Good or bad mouthing:** A node may perform a good or bad mouthing attack by giving a bad direct opinion towards a good node or by providing a good direct opinion towards a bad node. This hinders accurate trust evaluation.
- **Message modification:** A legitimate node with a symmetric key may modify MM. To prevent PI modification by other MCs, we use PI encryption keys as discussed in Section 2.1.
- **Packet dropping:** A node may drop packets based on its inherent selfish nature to lower service availability, leading to service unavailability and inaccurate trust evaluation.

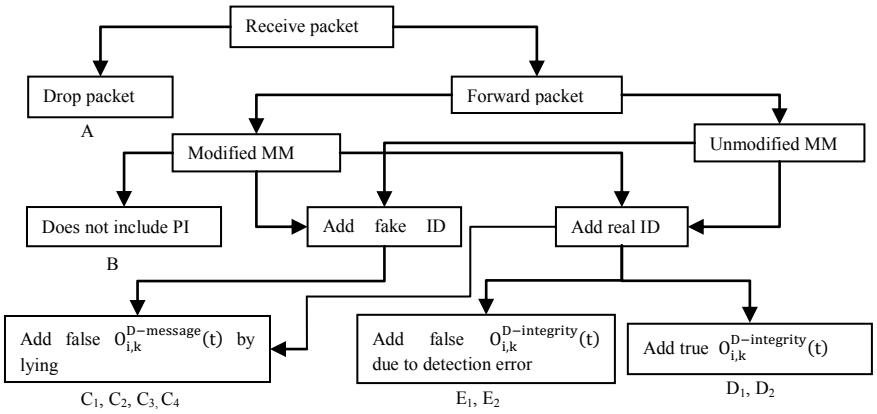

Fig. 1. Attack scenarios graph

Fig. 1 shows the attack scenarios considered in this work. Each node's behavior path is indicated with symbols such as A, B, C_1, C_2, C_3, C_4, D_1, D_2, E_1 and E_2. When a DN evaluates other nodes, if it does not see their ID in any received message and predicts energy depletion of the node, it will reduce a trust point for availability. If an attacker does not insert its ID or inserts a fake ID, it will be penalized by the decrease of the trust level. A smart attacker may want to reveal its real ID to avoid the penalty. If the attacker decides to insert a fake ID, it will provide false $O_{i,k}^{D-integrity}(t)$. Attacks can be performed with various combinations as shown in the paths described in Fig. 1.

2.3 Mobility Model and Node Deployment

We assume that nodes interact with each other not only to deliver messages, but also to exchange information for other purposes. A node is able to diagnose other nodes' attack behaviors based on its past direct experience. A given mission requires that each node, as a source, must send information to a list of destination nodes. Each node, as a DN, expects to receive information from a set of source nodes. For message delivery, nodes use the "store-and-forward" technique, meaning that a node carries messages until it encounters a MC.

Without loss of generality, we assume a square-shaped operational area consisting of m×m sub-grid areas with the width and height equal to wireless radio range (R). Initially nodes are randomly distributed over the operational area based on the uniform distribution. A node randomly moves to one of five locations (i.e., north, west, south, east, and current location) in accordance with its speed. The speed of node i, v_i, is chosen uniformly over (0, v_{max}] m/s where v_{max} is the maximum possible speed, and v_i is then fixed during the node's lifetime. The boundary grid areas are wrapped around (i.e., a torus is assumed) to avoid end effects. For simplicity, we assume that each node is located in the center of its sub-grid. Nodes are modeled with heterogeneous characteristics with different speed, energy level, monitoring capability (i.e., detection error), group join and leave rate, and cooperation probabilities (i.e., packet dropping), and honesty probabilities (i.e., good/bad mouthing, fake identity, message modification).

- **Speed (v_i)**: A node is assigned an average speed of its lifetime for analytical modeling, selecting from the range $(0, v_{max}]$ based on uniform distribution.
- **Energy level (E_j^{energy})**: A node is assigned an initial energy level selected from the range $[E_{min}, E_{max}]$ and its energy consumption is affected by its cooperativeness and membership status.
- **Detection error (P_i^{fp} / P_i^{fn})**: A node has monitoring capability with detection error probabilities of false positives and false negatives on integrity trust and predicting energy level for competence trust. Each node's detection error probabilities (P_i^{fp} and P_i^{fn}) are selected from the range $(0, P_{err}^{max}]$.
- **Group join and leave (λ / μ):** A node may leave or join a group where the inter-arrival time of the events is exponentially distributed with the rates λ and μ.
- **Cooperativeness (P_i^{coop}) and Integrity ($P_i^{integrity}$):** A node may drop a packet, or lie or modify a message based on the inherent characteristics of cooperativeness or integrity. We model these by assigning a seed probability for cooperativeness or integrity from the range $[GB_{min}, 1]$ based on uniform distribution.

2.4 Composite Trust Metric

The proposed trust metric consists of three trust properties: availability, integrity, and competence. First, ***availability*** property refers to service availability that is affected by system security and performance (e.g., quality-of-service). We mainly consider nodes' packet forwarding behavior to measure service availability. Loss of service availability may be caused by (1) a node's selfish or malicious behavior; (2) inherent network unreliability (i.e., link failure); (3) becoming a non-member by leaving the network; and (4) lack of access to a valid symmetric key. Second, ***integrity*** measures whether a node behaves without showing attacks described in Section 2.2. Third, ***competence*** property reflects the remaining battery lifetime of a node (a surrogate for resources available at the node) and the amount of positive experiences (PE).

Trust Aggregation: The trust value is formed with past evidence at time $(t - \Delta t)$ and new evidence, either direct or indirect, at time t. The trust value of node j evaluated by node i at time t is given by:

$$T_{i,j}(t) = T_{i,j}(t - \Delta t) + O_{i,j}^{new}(t), \quad T_{i,j}(t = 0) = \lceil (TV_{min} + TV_{max})/2 \rceil \qquad (2)$$

A trust value is a real number and clipped into the range $[TV_{min,}\, TV_{max}]$. The initial trust value $T_{i,j}(t = 0)$ is at the midpoint of the allowed range. Notice that the overall trust $T_{i,j}(t)$ is updated based on new direct or indirect observations on top of past experience at t-Δt.

Trust Formation: Newly observed (either directly or indirectly) trust evidence comprises three trust properties:

$$O_{i,j}^{new}(t) = O_{i,j}^{availability}(t) + O_{i,j}^{integrity}(t) + O_{i,j}^{competence}(t) \qquad (3)$$

When nodes i and j encounter each other as 1-hop neighbors, node i will entirely rely on direct observations towards node j's behaviors to collect new evidence at time t. Direct trust evaluation can be assessed between any two encountering nodes based on their own assessment capability. Availability is measured by whether a node is available to serve requests. Integrity is evaluated based on three attack behaviors. Competence is assessed by energy level and positive experience. Thus, each trust component is evaluated with a single observation or multiple observations where each observation is counted as equal. We discuss details of the three trust components below in Direct Trust Evidences.

Note that node i is not necessarily a DN. However, indirect trust evaluation can be only conducted when node i is a DN. That is, node i (DN) will rely on received messages to evaluate trustworthiness of node j. In this case, time t represents the time that the DN evaluates trust towards node j based on the received information even if the trust evidences are collected by intermediate MCs on the way to deliver the message to the DN.

Direct Trust Evidences: When nodes i and j are 1-hop neighbors, a trust value is computed based only on direct new observations plus past experiences. Recall that nodes interact with each other for other purposes and are able to leverage the experience to assess direct trust towards 1-hop neighbors. We define α as a reward or penalty unit in trust level for each trust property.

- **Direct availability ($O_{i,j}^{D-availability}(t)$):** This is α if node i has received message(s) from its 1-hop neighbor node j during the last Δt period; -α otherwise. $O_{i,j}^{D-availability}(t)$ lies in $[-\alpha, \alpha]$.
- **Direct integrity ($O_{i,j}^{D-integrity}(t)$):** This consists of three trust components:

$$O_{i,j}^{D-integrity}(t) = O_{i,j}^{D-message}(t) + O_{i,j}^{D-honesty}(t) + O_{i,j}^{D-identity}(t) \qquad (4)$$

$O_{i,j}^{D-message}(t)$ is α if node i believes node j did not modify MM; - α otherwise. $O_{i,j}^{D-honesty}(t)$ is α if node i believes node j did not lie about the integrity of the

previous MC; - α otherwise. $O_{i,j}^{D-identity}(t)$ is α if node i believes node j inserted a real ID; - α otherwise. $O_{i,j}^{D-integrity}(t)$ lies in $[-3\alpha, 3\alpha]$.

- **Direct competence ($\mathbf{O_{i,j}^{D-competence}(t)}$):** This is formed with two trust components, energy level and positive experience:

$$O_{i,j}^{D-competence}(t) = O_{i,j}^{D-energy}(t) + O_{i,j}^{D-PE}(t) \quad (5)$$

$O_{i,j}^{D-competence}(t)$ is 0 when $O_{i,j}^{D-availability}(t) < 0$. $O_{i,j}^{D-energy}(t)$ is α if $E_j^{energy}(t) > E^{th}$ where E^{th} is the minimum energy threshold required to execute a mission; - α otherwise. $O_{i,j}^{D-PE}(t)$ is α if $O_{i,j}^{D-availability}(t) + O_{i,j}^{D-integrity}(t) == 4\alpha$, meaning that node j gains extra reward when it behaves perfectly in both availability and integrity; - α otherwise. $E_j^{energy}(t)$ is extrapolated based on the direct (prior knowledge on initial energy level) and indirect information (availability). Note that we consider probabilities of false positives and negatives (P_i^{fp} and P_i^{fn}) in the above direct trust evaluation for an imperfect monitoring mechanism installed in each node. Imperfect detection is applied for integrity trust and energy level. Availability trust depends upon a receipt of the packet. Positive experience in competence trust is evaluated through the three components of integrity trust evaluated by considering detection errors and availability trust. $O_{i,j}^{D-competence}(t)$ ranges over $[-2\alpha, 2\alpha]$.

Indirect Trust Evidences: When node j is more than 1-hop distant from node i, node i (DN) will rely on provenance information in a received message, if any, to evaluate node j. However, node i may not receive any messages enclosing node j's ID (even no-ID insertion attack is not caught). In this case, when the energy level of node j is predicted as depleted, the following penalty will be given:

$$O_{i,j}^{ID-availability}(t) = -\alpha\,,\ O_{i,j}^{ID-integrity}(t) = O_{i,j}^{ID-competence}(t) = 0 \quad (6)$$

If the node is caught by a DN for no-ID insertion or no-PI insertion, then it will be penalized for unavailability in addition to the ID attack as well.

- **Indirect availability ($\mathbf{O_{i,j}^{ID-availability}(t)}$):** When node i, as a DN, receives a message, it evaluates node j's availability as follows:

$$O_{i,j}^{ID-availability}(t) = \begin{cases} \alpha \text{ if } O_{i,j}^{ID-identity}(t) > 0; \\ 0 \text{ otherwise;} \end{cases} \quad (7)$$

When node j's ID is shown in the received message and proven to be authentic, node j's availability trust is incremented by α. If node j's ID is inserted by a fake identity attacker, node j will not be penalized. See Equation 9 for $O_{i,j}^{ID-identity}(t)$.

- **Indirect integrity ($\mathbf{O_{i,j}^{ID-integrity}(t)}$):** Similar with direct integrity trust, this is formed with three components as follows:

$$O_{i,j}^{ID-integrity}(t) = O_{i,j}^{ID-message}(t) + O_{i,j}^{ID-honesty}(t) + O_{i,j}^{ID-identity}(t) \quad (8)$$

Indirect identity trust, $O_{i,j}^{ID-identity}(t)$, is computed by:

$$O_{i,j}^{ID-identity}(t) = \begin{cases} \alpha \text{ if } D(ID_j, preID(m(j))) == 0 \text{ and } O_{m(j),j}^{D-identity}(t) > 0; \\ 0 \text{ otherwise;} \end{cases} \quad (9)$$

$$O_{i,k}^{ID-identity}(t) = \begin{cases} -\alpha \text{ if } O_{i,j}^{ID-identity}(t) == 0; \\ 0 \text{ otherwise;} \end{cases}$$

here m (j) indicates the next MC to node j and $O_{m(j),j}^{D-identity}(t)$ is only considered when $T_{i,m(j)}(t-\Delta t) \geq T_{min}$, implying only trustworthy nodes' information is evaluated. $D(ID_j, preID(m(j)))$ returns 0 when the two IDs are the same; 1 otherwise. ID_j is the ID inserted by node j and $preID(m(j))$ is the previous MC's ID provided by node $m(j)$. $O_{m(j),j}^{D-identity}(t)$ is the direct observation on identity trust towards node j by the next MC $m(j)$. When node j's ID is proven to be true based on Equation 9, node j's identity trust is incremented by α. Otherwise, node j is not penalized since it is a victim due to a fake identity attack performed by another node. If caught, the fake identity attacker, node k, is penalized instead.

Indirect honesty trust, $O_{i,j}^{ID-honesty}(t)$, is obtained by:

$$O_{i,j}^{ID-honesty}(t) = \begin{cases} \alpha \text{ if } T_{i,j}(t-\Delta t) \geq T_{min} \text{ and } O_{m(j),j}^{D-honesty}(t) > 0; \\ -\alpha \text{ otherwise;} \end{cases} \quad (10)$$

Similarly, $m(j)$ is the next MC to node j and $O_{m(j),j}^{D-honesty}$ is only evaluated when $T_{i,m(j)}(t-\Delta t) \geq T_{min}$. Note that direct evidences used are collected when a message travels through intermediate MCs. At time t, node i (DN) evaluates node j based on the direct evidence provided by node $m(j)$, the next MC of node j.

$O_{i,j}^{ID-message}(t)$ is evaluated based on the other two integrity trust components (identity and honesty) and a direct opinion of the next MC $m(j)$ towards the previous MC j on mission message modification, and computed by:

$$O_{i,j}^{ID-message}(t) = \begin{cases} \alpha \text{ if } O_{i,j}^{ID-identity}(t) > 0 \text{ and } O_{i,j}^{ID-honesty}(t) > 0 \text{ and } O_{m(j),j}^{D-message}(t) > 0; \\ -\alpha \text{ otherwise;} \end{cases} \quad (11)$$

$O_{m(j),j}^{D-message}(t)$ is a direct message trust opinion of the next MC $m(j)$ towards the previous MC j where $m(j)$ has the past trust level, $T_{i,m(j)}(t-\Delta t) \geq T_{min}$.

- **Indirect competence ($O_{i,j}^{ID-competence}(t)$):** This is measured similarly as direct competence, but based on indirect evidences. This is given by:

$$O_{i,j}^{ID-competence}(t) = O_{i,j}^{energy}(t) + O_{i,j}^{ID-PE}(t) \quad (12)$$

2.5 Metrics

Recall that our goal in developing the proposed provenance model was to estimate trust accurately and efficiently. We use two performance metrics to evaluate the proposed trust model as follows:

- **Trust Bias ($T_{i,j}^{bias}$):** This is the time-averaged difference between trust of node j evaluated by node i and objective trust of node j evaluated by all encountered nodes based on direct observations with no detection errors. This metric considers both false positives and negatives. $T_{i,j}(t)$ is the trust value of node j evaluated by node i at time t and $OT_j(t)$ is an objective trust value of node j based on aggregated direct observations of all encountered nodes at time t. Given the entire mission lifetime LT, $T_{i,j}^{bias}$ is obtained by:

$$T_{i,j}^{bias} = \frac{\int_0^{LT} T_{i,j}^{bias}(t)}{LT} \quad \text{where } T_{i,j}^{bias}(t) = |T_{i,j}(t) - OT_j(t)|/OT_j(t) \tag{13}$$

- **Communication Overhead (C_{total}):** This is the communication cost per time unit (sec.) for a node to deal with trust evaluation ($C_{TE}(t)$) and message delivery ($C_{MD}(t)$) during the entire mission lifetime, LT. C_{total} is computed by:

$$C_{total} = \frac{\int_0^{LT}(C_{TE}(t) + C_{MD}(t))}{LT} \tag{14}$$

- **Mission Message Correctness (N_{CR}):** This refers to how many packets a DN receives correctly during the entire mission lifetime, LT. The trustworthiness of intermediate MCs significantly affects the correctness of received messages. This is computed by:

$$N_{CR} = \sum_{p \in P} \prod_{k \in L_p} P_k^{p-message}(t) \tag{15}$$

$$P_k^{p-message}(t) = \begin{cases} 1 \text{ if a MC k did not modify message p;} \\ 0 \text{ otherwise;} \end{cases}$$

here P is the set of messages sent by a source node to a DN and the k nodes are intermediate MCs delivering message p. L_p is the set of all intermediate MCs involved in delivering each message p.

3 Hierarchical Modeling using Stochastic Petri Nets

We use SPN because of its efficient representations of a large number of states where the underlying model is a continuous-time Markov or semi-Markov chain. We develop a hierarchical modeling technique based on SPN to avoid state explosion problems and to improve solution efficiency for realizing and describing the behaviors of each node and obtaining objective trust values.

We develop event subnets to describe a node's behavior and its actual trust value as shown in Fig. 2. A hierarchical SPN technique is used to derive interactions or trust relationships with other nodes in the system. We conduct this process by running the SPN subnet N times for the N nodes in the network. We use the information obtained from SPN for trust evaluation. In SPN, we call each oval shown in Fig. 2 a "place" where "mark (place name)" is the number of tokens in the place. The number of tokens in different places indicates the status (state) of a node. Each transition bar (i.e., T_NAME) is the rate at which the corresponding event is triggered.

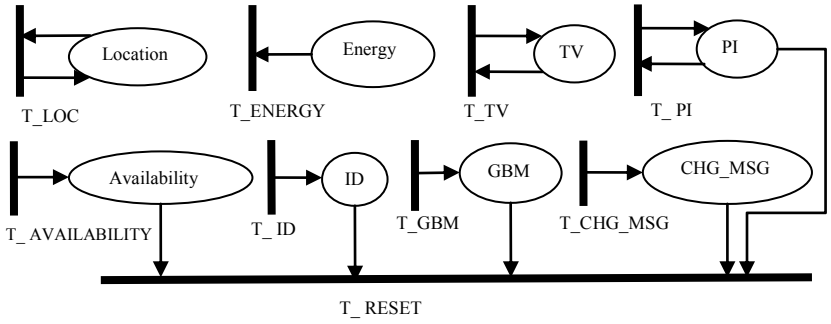

Fig. 2. Node SPN Subnet

Location Subnet: This subnet computes the probability that node i is in a particular grid area j at time t. This information along with the information of other nodes' locations at time t provides the information about when two nodes encounter as 1-hop neighbors at time t. Since node movements are assumed to be independent, the probability that two nodes are in a particular location at time t is given by the product of the two individual probabilities. Location probabilities are used to compute the probabilities that two nodes encounter or a node obtains a valid symmetric key based on the location of itself and TA's fixed location. The transition T_LOC rate is computed as v_i/R where v_i is node i's average speed given and R is radio range.

Energy Subnet: This subnet is used to obtain each node's energy lifetime. The number of tokens in place Energy indicates the battery life (hours) in energy. We approximately estimate energy consumption depending on a node's status: available vs. unavailable. We regard a node's availability as forwarding packets where a node may drop packets with any reason (See Availability Subnet below). When a node is not available, energy consumption is slowed down. The transition T_ENERGY is modeled by:

$$\text{if(mark(Availability)} == 0),\ \text{rate(T\_ENERGY)} = 1/(2T_{\text{energy}}) \qquad (16)$$

$$\text{else rate(T\_ENERGY)} = 1/T_{\text{energy}}$$

We assume that one token represents energy consumed for T_{energy} for normal activities. When a node is in sleep mode or does not serve any request (i.e., unavailable status), it is predicted as consuming one half of normal energy consumption.

Trust Value Subnet: The number of tokens in place TV represents a direct trust value observed by 1-hop neighbors. We assume that a node shows consistent behavior patterns to all nodes, so the views of 1-hop neighbors towards the same node are assumed synchronized. Thus, mark (TV) is computed based on the equations on direct trust evidences described in Section 2.4 without considering any detection error. This trust value is used as an objective trust to obtain a trust value at time t based on direct observations by all encountered nodes. Direct trust evaluation is performed per encounter interval with the transition T_TV rate being $1/T_{i,\text{encounter}}$, meaning that node i encounters another node with the average inter-arrival time of $T_{i,\text{encounter}}$. $T_{i,\text{encounter}}$ is computed by $\sum_{j\in N} R/\left[\left(P_i^{loc-l}\sum_{n\in S} P_j^{loc-n}\right)(v_i + v_j)\right]$ where node j

belongs to the set N including all nodes in the network, R is radio range, P_i^{loc-l} is the time-averaged probability that node i is located in area l, S is the set of adjacent locations of node i, and v_i is the speed of node i.

Availability Subnet: A token in place Availability indicates that node i is available and cooperative upon receiving a request; zero token otherwise. The rate for the transition T_AVAILABILITY is affected by: (1) join probability (i.e., $P^{\lambda} = \lambda/(\lambda + \mu)$ where λ and μ are join and leave rates); (2) whether node i is able to obtain a symmetric key from the closest TA (P_{sk}); (3) the probability that node i is cooperative to serve packet forwarding (P_i^{coop}); (4) link reliability based on network or node conditions (P_{link}); and (5) whether or not a node's trust is below T_{min} ($P_{untrustworthy}$). Upon the receipt of a newly arrived packet, a node may become available as determined by the following condition:

$$\text{if}\big(\text{mark(Availability)} == 0 \text{ and } P_{sk} > 0 \text{ and } P_{untrustworthy} > 0\big) \quad (17)$$

$$\text{rate(T\_AVAILABILITY)} = \big(P^{\lambda} P_i^{coop} P_{link}\big)/T_{i,encounter}$$

$$\text{else} \quad \text{disable T\_AVAILABILITY}$$

P_{sk} is computed based on node i's location and fixed locations of TAs. P_{sk} is 1 when a symmetric key is obtainable; 0 otherwise. $P_{untrustworthy}$ is 1 when a node has its trust value below T_{min}.

PI Subnet: A token in place PI means that node i decides to insert provenance information; no token otherwise. The rate for transition T_PI is given by $P_i^{coop}/T_{i,encounter}$.

ID, GBM and CHG_MSG Subnets: Identity, message, and honesty trust components in integrity are evaluated similarly. When place ID, GBM or CHG_MSG has a token, it means that a respective attack is performed; zero token otherwise. The rates for transitions T_ID, T_GBM and T_CHG_MSG are given by $(1 - P_i^{integrity})/T_{i,encounter}$. These attacks do not occur when no provenance information is inserted, i.e., mark(PI)==0. A good or bad mouthing attack occurs when a fake ID is inserted.

Transition T_RESET flushes all tokens from those places with output arcs into the transition upon encountering a new node with the rate of $1/T_{i,encounter}$.

4 Numerical Analysis and Results

This section compares the proposed provenance-based trust model (PT) with a baseline trust model (BT) in terms of the proposed metrics. We choose the model described in our prior work [2] as the existing BT that evaluates a node's trust based on direct observation or experience and recommendations. For fair comparison, we slightly modify BT that fits the trust metric considered in this work. BT uses the same trust metric as PT except the way it aggregates trust with direct and indirect trust evidences based on recommendations as follows:

$$T_{i,j}(t) = T_{i,j}(t - \Delta t) + \beta\, O_{i,j}^{direct-new}(t) + (1 - \beta)\, O_{i,j}^{indirect-new}(t) \quad (18)$$

where $O_{i,j}^{direct-new}(t)$ is computed based on Equation 3 and β and (1-β) are the weights applied to direct and indirect trust evidences. $O_{i,j}^{indirect-new}(t)$ is evaluated by recommendations from all encountered nodes. The encountered nodes pass recommendations only based on direct observation in order to avoid any security vulnerability by passing a derived trust.

Table 1. Default values used

Parameter	Value	Parameter	Value
v_{max}	15 m/sec.	β	0.8
GB_{min}	0.8	E^{th}	0
T_{min}	5, 10, 15, 20, 25	LT	100,000 sec.
α	1	P_{link}	0.99
R	100 m	$[TV_{min}, TV_{max}]$	[0, 30]
λ	Once per hour	μ	Once per 4 hours
P_{error}^{max}	0.01	$[E_{min}, E_{max}]$	[12, 24]

In this case study, 165 packets each with 2 copies (total 330 packets) are sent from a source to a destination. In each run, 20 different source-destination pairs are deployed. We pick one pair and show the results (source: node 3, destination: node 15). A total of 20 nodes are spread over the operational area divided into 6 × 6 regions. The results are shown with the average values computed over 100 runs of trust evaluation.

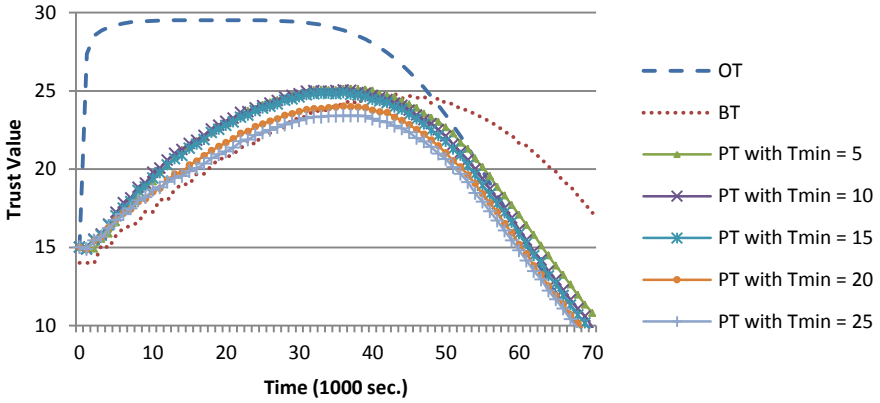

Fig. 3. Trust values over time: OT vs. BT vs. PT

Fig. 3 shows the average trust values of all nodes evaluated by a DN over time in OT, BT, and PT with various T_{min} based on Equation 13. OT ($OT_j(t)$ in Equation 13) is the objective trust value based on only direct trust evaluation by all encountered nodes. BT is not affected by using different T_{min} since trust evaluation is not dependent upon the selection of the next MC in the message delivery. Thus, we show only one curve under BT. However, PT is affected by various T_{min} used since provenance information tagged in the main message is used as indirect trust evidences

for overall trust evaluation. BT underestimates trust values in the beginning half while overestimating trust values in the rest of the mission lifetime. Overall, PT performs better than BT without underestimating trust values in the beginning and showing relatively accurate trust assessment in the end. While BT only depends on the encounter event where nodes i and j can exchange information, PT can collect trust evidences indirectly based on the provenance information tagged with main messages, leading to better trust accuracy. As T_{min} increases, PT further underestimates trust values because using higher T_{min} in selecting the next MC only updates trust values of highly trustworthy nodes while decaying those of less trustworthy nodes due to their unavailability.

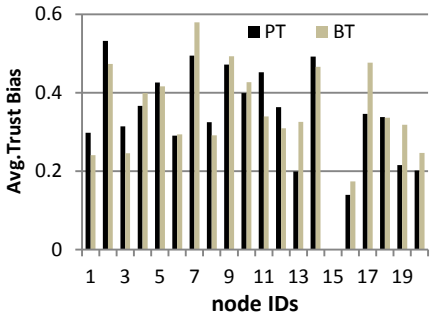

Fig. 4. Average trust bias per node: PT vs. BT

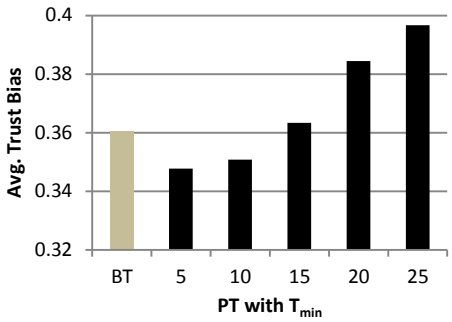

Fig. 5. Average trust bias of all nodes: PT vs. BT

Figs. 4 and 5 (computed based on Equation 13) confirm the observation and conclusion derived from Fig. 3. Fig. 4 is the time-averaged trust bias per node. Fig. 5 is the overall time-averaged trust bias of all nodes. PT performs significantly better than BT in trust accuracy when a lower T_{min} is used.

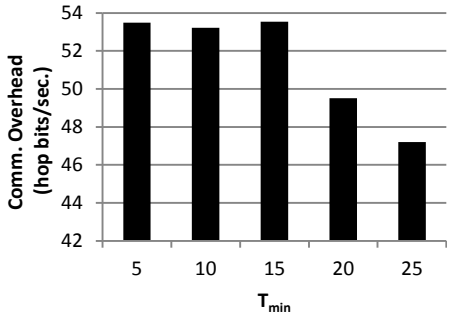

Fig. 6. Communication overhead in PT vs. system trust threshold (T_{min})

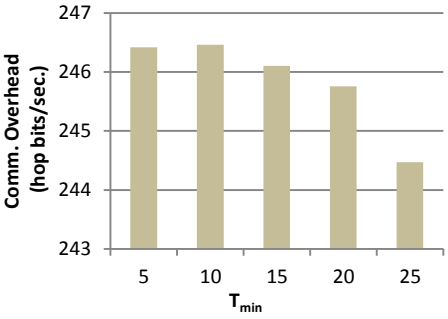

Fig. 7. Communication overhead in BT vs. system trust threshold (T_{min})

Figs. 6 and 7 show communication overhead (C_{total}) under PT and BT with respect to various T_{min} values based on Equation 14. When a higher T_{min} is used, a lower C_{total} results due to a smaller number of nodes with high enough trust values to do message delivery. The average C_{total} over different T_{min} in BT is 245.84 hop bits/sec. while that in PT is 51.39 hop bits/sec. This demonstrates that PT significantly reduces communication overhead compared to BT while achieving better performance in trust accuracy with low $T_{min} < 15$, as shown in Figs. 3, 4, and 5.

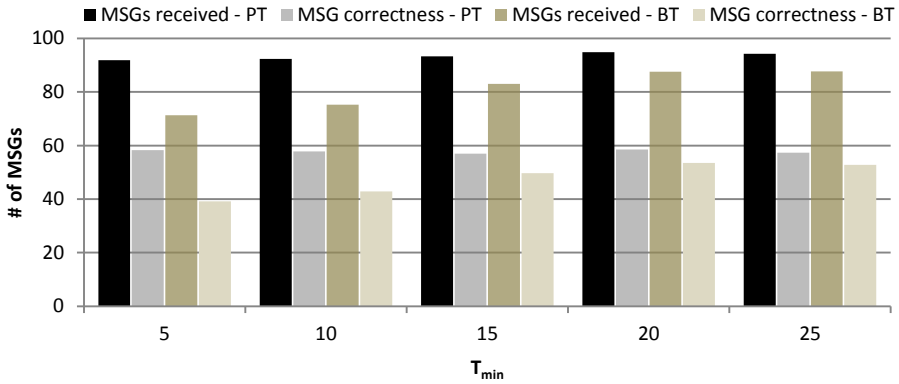

Fig. 8. The numbers of messages received and messages received correctly: PT vs. BT

Fig. 8 compares the two schemes in terms of the number of messages received and the number of correct messages among the received messages. In BT, as a higher T_{min} is used, more messages are received and more messages are correct among the received messages (computed based on Equation 15). That is, selecting a highly trustworthy node as the next MC positively affects message delivery ratio as well as message correctness. In PT, we do not observe much sensitivity over different T_{min} in terms of these two metrics. This is because trust update in PT is affected by whether a received message has provenance information about each node. This is determined by which node is selected as the next MC using T_{min}. When a higher T_{min} is used, only the trust values of nodes with higher trust values are updated while the trust values of other nodes with lower trust values decay over time due to unavailability, since they are not being selected as the next MC. Thus, the benefit of using a higher T_{min} is not prominent because nodes with the trust value above T_{min} may not be found easily with high T_{min}. In addition, since PT tends to underestimate trust values of nodes, it selects a more qualified node as the next MC than what is required, thus lowering risk. On the other hand, BT is more likely to overestimate especially towards the end of mission lifetime. This leads to a next MC with a less qualified node than what is expected, thus increasing risk. Therefore, overall PT performs better than BT.

5 Conclusions and Future Work

This paper proposed a provenance-based trust model that achieves better trust accuracy compared to an existing scheme while significantly reducing communication overhead for trust evaluation. The proposed scheme outperformed the existing scheme in three metrics: trust accuracy, communication overhead, and the number of messages received and message correctness.

We plan to extend this work by: conducting further sensitivity analysis; refining our attack model; and introducing dynamic minimum trust thresholds.

Acknowledgement

Dr. MoonJeong Chang was supported in part by the Army Research Office under Grant W911NF-12-1-0016.

References

1. S.M.I. Alam and S. Fahmy, "Energy-efficient provenance transmission in large-scale wireless sensor networks," IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks, Lucca, Italy, June 20-23, 2011.
2. I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust management for encounter-based touting in delay tolerant networks," IEEE Global Telecommunications Conf., pp. 1-6, Miami, FL, Dec. 6-10, 2010.
3. J.H. Cho, A. Swami, and I.R. Chen, "A survey of trust management in mobile ad hoc networks," IEEE Communications Surveys and Tutorials, vol. 13, no. 4, pp. 562-583, 2011.
4. C. Dai, D9. Lin, E. Bertino, and M. Kantarcioglu, "An approach to evaluate data trustworthiness based on data provenance," Proc. 5th VLDB Workshop on Secure Data Management, LNCS, vol. 5159, pp. 82–98, Auckland, New Zealand, Aug. 24, 2008.
5. J. Freire, D. Koop, E. Santos, and C.T. Silva, "Provenance for computational tasks: A survey," IEEE Computing in Science and Engineering, vol. 10, no. 3, pp. 11-21, 2008.
6. J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering," Provenance and Annotation of Data, LNCS, vol. 4145, pp. 101-108, 2006.
7. H.S. Lim, Y.-S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," Proc. 7th Int'l Workshop on Data Management for Sensor Networks, Singapore, Singapore, Sept. 13, 2010.
8. Y. Liu, J. Futrelle, J. Myers, A. Rodriguez, and R. Kooper, "A provenance-aware virtual sensor system using the open provenance model," Int'l Symposium on Collaborative Technologies and Systems, pp. 330-339, Chicago, IL, May 17-21, 2010.
9. P. McDaniel, "Data provenance and security," IEEE Security and Privacy, vol. 9, no. 2, pp. 83-85, 2011.
10. L. Moreau, J. Freire, J. Futrelle, R.E. McGrath, J. Myers, and P. Paulson, "The open provenance model: an overview," Int'l Provenance and Annotation Workshop, Salt Lake City, Utah, LNCS 5272, pp. 323–326, June 17-18, 2008.
11. A. Perrig and J.D. Tygar, Secure Broadcast Communication in Wired and Wireless Networks, Kluwer Academic Publishers, 2002.
12. S. Rajbhandari, I. Wootten, A.S. Ali, and O.F. Rana, "Evaluating provenance-based trust for scientific workflows," 6th IEEE Int'l Symposium on Cluster Computing and the Grid, vol. 1, pp. 365-372, Singapore, May 16-19, 2006.
13. T. Spyropoulos, R.N. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," Wireless Networks, vol. 16, no. 8, pp. 2349–2370, 2010.
14. M. Srivatsa, W. Gao, and A. Iyengar, "Provenance-driven data dissemination in disruption tolerant networks," Proc. 14th Int'l Conf. on Information Fusion, Chicago, IL, July 5-8, 2011.
15. S. Sultana, E. Bertino, and M. Shehab, "A provenance based mechanism to identify malicious packet dropping adversaries in sensor networks," 31st Int'l Conf. on Distributed Computing Systems Workshops, pp. 332-338, Minneapolis, MN, June 20-24, 2011.
16. X. Wang, K. Govindan, and P. Mohapatra, "Collusion-resilient quality of information evaluation based on information provenance," 8th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 395 - 403, Salt Lake City, Utah, June 27-30, 2011.
17. B. Yu, S. Kallurkar, and R. Flo, "A Demspter-Shafer approach to provenance-aware trust assessment," Int'l Symposium on Collaborative Technologies and Systems, pp. 383-390, Irvine, CA, May 29-23, 2008.
18. National Cyber Security Research and Development Challenges: Related to Economics, Physical Infrastructure and Human Behavior, An Industry, Academic and Government Perspective, 2009.